E. Help Desk : A NATURAL LANGUAGE PROCESSING FOR ANSWERING QUERY

By

Norwin Shariman B. Mohd. Nor
WEK 990162

Supervisor

PROF. MADYA DR. SAPIYAN BABA

A Final Year Project, submitted as a partial requirement for the degree of Computer Science, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur. 2000/2001

ABSTRACT

This report is about a search engine, E. Help Desk that is to be deployed in a location where usually the users have a brief contact with the search engine. Thus it is designed to be very easy to use as it is designed to be intuitive, so as the learning curve for the search engine is very shallow.

The system is divided into two modules, the user interface and the administrator module. The user interface provides a basic search engine capability to the user to enable them to search for relevant article. In regard to place like a shopping complex, for instance, the user can search for shops and public amnesties.

The administrator module provides the capability for the people involved in the maintenance to update the information. This will be very useful in some application where the information changes often.

The designing of the system uses the software engineering methodology to enable the designing of a system that are relatively trouble free. Thus, the authour tries to make sure that every aspect in the development be planned.

ACKNOWLEDGEMENTS

I am very grateful to a number of people who have helped contribute to in my completing this Final Year Project. I would like to express my gratitude to my supervisor, Dr Sapiyan Baba for his patience and guidance during the months of my project work. He set aside his time in order to accommodate to my meetings regarding this project. I would also like to take this opportunity to thank Dr Woo Chaw Seng for being my moderator for this project.

Besides that, special thanks also to my family members who have always given me utmost moral support while I was trying to complete the system. I also wanted to express my gratitude to Bustaman, whose help have saved me from a particular problem. I am also really grateful to all my friends who have tried the system prototypes and gave me feedback to improve my system.

Thank you.

Table Of Contents

CHAPTER 1
INTRODUCTION

1.1 Introduction

The search engine's role

Because of the growth in information management, many techniques and methods have been devised to manage information and to answer queries from the users. Unfortunately, while there is a great advancement in the business based information management, there has been only scarce methods have been implemented to make the end users' interaction better.

What is the importance of a natural language search engine?

Although by itself, a search engine is simple in a sense that in only search for the answers that the users have keyed in, we can't deny that the search engine has become the common tool for extracting information from sources that sometimes impossible to shift through without the use of it.

What is E. Help Desk?

E. Help Desk is a search engine used for kiosk-based deployment. The difference between it and the normal search engine is that is supports a natural language query to a limited degree, not just by searching all through the index using the given keywords.

1.2 Objectives

Among the objectives that are to be achieved with the development and implementation of E. Help Desk are as follows:

- To design and develop an efficient search engine that will be very easy to use, so much so that it hardly needs help files. This "intuitive" interface is better suited to the use in a location where the interaction time between the users and the system is quite short.
- To design a modular search engine that can be used with other components.
- To design a search engine that can also be deployed various places

1.3 Scope of project

E. Help Desk is designed divided into two modules which are the user interface module and the administrator module. However, just one module can be deployed, which is the user interface module. The administrator module contains module for updating information.

1.3.1. The public user interface

The public user interface of the system has the search engine where the user only have to key in their query so as to retrieve the probable answers to their query.

1.3.2 Administrator Interface Module

People who are involved in maintaining the system will use the administrator module. The maintenance includes the facility to update information, i.e. add new information to the knowledge base so that the database will be up to date. But, this depends on the

deployment. A basic system deployed in a shopping complex will not have this necessity as the information used may not normally change, such as the information on tenants or the location of the basic amnesties in the building.

1.4 Significance

1. The E. Help Desk will help the user to find answers using simple language queries.
2. The E. Help Desk will also be able to show some extra information, such as the location of the place queried if applicable

CHAPTER 2

REVIEW OF LITERATURE

2.1 Purpose

1. To gather information about the system we intend to develop.

2. Evaluate existing systems on the same topic so that a better product can be developed.

3. Comparison of a few software, tools and approaches is important for the best outcome. Without this analysis, we would not be able to identify the strengths and weaknesses.

2.2 Approach

Information Gathering

In order to attain information to build the system there a few main resources that I took into account. This part of the proposal is very important for further development of the system therefore it involved a lot of research. Information sought for this project is from four main resources, are described as follows:-

2.2.1 Search engine

Using search engine to search through the internet, we can retrieve information regarding many concepts and things that can be exhaustive, if the search was done correctly. The author used the search engine to gather information on systems that are based on Natural Language Processing.

www.google.com
www.yahoo.com
www.infoseek.com
www.altavista.com
www.excite.com

Mainly, all these search engines were used to:-

1. ***Find out the existing natural language processing system whose information are available for public scrutiny.***

These systems are evaluated based upon the features they have and how user friendly they are. Key words that are used for this purpose are words that are relevant to the natural language processing, such as 'natural language processing', and 'parsing'

2. ***Find out the software available for web development***

The limitations and strengths of most development software are considered for the selection of software for the development the authour's system. The main keywords used were application development such as , Visual Prolog and Microsoft Visual C++. Some information regarding ready to deploy systems where only information need to be supplied is also examined.

3. Find out the programming tools for development.

For each of the software development platform, tools that are available for free were examined to determine whether they are suitable for use in the systems.

4. Find out good methods of designing the application on each platform

In developing the system on any particular platform, it is imperative to find out the best way do it so that it doesn't cause any problem when deployed on the platform.

2.2.2 Library resources

Library resources, mainly books were found from the main library of University Malaya. Books on web development software were read to put more strength on the findings in the net.

Besides that, dictionaries available in the library were helpful in the sense that they provide a clearer picture of arranging the data in a more meaningful manner. Since E. Help Desk is a search engine, more stress was given on this field when looking for appropriate examples.

System Analysis and Design books helped to understand the methodology for designing a system. These books were helpful also in the designing phase of the system such as creating data flow diagram (DFD), flow chart and Gantt chart.

Some of the web development books in the library were useful in order to gain knowledge in creating a more meaningful and professional-looking web site.

Understanding of database systems is important for developing E. Help Desk. Therefore books on database applications were read and analysed.

2.2.3 Bilik Dokumen

Another source for gathering information is the Bilik Dokumen in the Faculty of Computer Science and Information Technology. In the Bilik Dokumen, the authour have made references to some projects and documentations by the seniors and some of my peers, which are relevant to my project. I manage to analyse a few search engines.

Some of the reports were useful in the sense that they provide information regarding the software used and the tools needed for any particular software.

Besides that, the documentation gave a brief idea on how to go about in developing a system from scratch, which is very important at the beginning stage of developing a new system.

The resource in Bilik Dokumen consists of reports on several different kind of applications including the web-based ones. So, this gave the authour a chance to compare a few projects, which the authour used as the best approach for developing E. Help Desk.

2.2.4 Interviewing

An information-gathering interview is a directed conversation with a specific purpose that uses a question-and-answer format. The five major steps in interview preparation are shown in Figure 5.2. These steps include a range of activities from gathering basic background material to deciding whom to interview.

Steps in planning the interview :-

1. ***Read Background Material***
2. ***Establish Interviewing Objectives***
3. ***Decide Whom to Interview***
4. ***Prepare the Interviewee***
5. ***Decide on Question Types***

A few interview sessions were done with the dealers from e few second-hand car companies and users, mainly students and my relatives.

2.3 Findings

2.3.1 Search engine

2.3.2 Books

1. System Analysis and Design, 1979

This text combines concept and practice in one volume and gives a broad yet specific treatment of the makeup, analysis, design, and implementation of systems projects. It uses a practical approach to real-life situations based on the author's years of experience in the field.

2. System Analysis and Design, 1999

This book emphasizes on the designing phase of a system in detail, the methodologies and teaches the proper way to write a proposal.

3. Software Engineering, A Practioner's Approach, 1992

This text consists the method to design a system. For example, the architectural design, functional and non-functional functions of a system.

2.3.3. Bilik Dokumen

Online Dictionary: ODOCT

An online dictionary on computer terms, which the authour used as reference in report writing. This system is developed by one of the Faculty of Computer Science and Information Technology's student.

2.4. Analysis

2.4.1. Designing An Application

General guidelines for designing Application

1. Use appropriate professional tools

The usage of programming tools such as Microsoft's Visual C++ 6.0 or Visual Prolog will definitely assist in the designing the application.

2. Study Other Applications

Generally there are two methods of designing an application. The first one is to design the application so simple that it doesn't need any help to be implemented. This is called 'intuitive application'. Another one is the general development policy where help were provided as needed. Each of these is suited to some applications only, while being ineffective in others.

3. Plan Ahead

In designing, it is always better to plan ahead so that satisfactory results could be attained.

Good screen design

There are four guidelines for a good screen design.

- **Keep the screen simple**
- **Create an attractive screen**

2.4.2. Software

Software Evaluation

Software evaluation is done based on all these factors. (*Kendall & Kendall, 1999*)

- **performance effectiveness**

Able to perform all required tasks

Able to perform all tasks that may be desired at some time in the future

Well design display screens

Adequate capacity

- **performance efficiency**

Fast response time

Efficient input

Efficient output

Efficient storage data

Efficient back-up

► **easy to use**

Satisfactory user interface

Help menus available

Readme files for last minute changes

Flexible interface

Adequate feedback

Good error recovery

► **flexibility**

Options for input

Options for output

Usable with other software

► **quality of documentation**

Good Organization

Adequate online tutorial

Web site with FAQ

► **manufacturer support**

Technology support hotline

Newsletter/email

Web site with downloadable product updates

Benefits and Limitations of the software considered.

1. Microsoft Visual C++ 6.0

Microsoft Visual C++ 6.0 includes a number of technical breakthroughs that provide users a fast and easy way to create and manage applications

New Features

- easier installation
- easier graphical user interface development
- compartmentalized (object oriented) software development platform
- rapid Windows-based application development

Benefits

The software is one of the leading Microsoft Visual-based application development platform, which is used in conjunction with other Microsoft Visual Studio platform components, such as Visual Basic 6.0 and Fox-Pro.

The usage of Microsoft Foundation Class ensure that with proper coding and implementation, the Windows based application that developed on the platform is quite robust.

Limitation

Unfortunately, being primarily C++ based software implementation, it doesn't function very well as a Natural Language Processing system. Thus, although it have, as its advantage, ease of use, this limitation cancels out the benefit.

2. Visual Prolog *(http://pdc.de.uk)*

New Features

- easier integration with Windows
- easier graphical user interface development
- better Windows-based application development using Prolog

Benefits

The software is another implementation of Prolog language. Thus, it more suited with the Artificial Intelligence based application, such as Natural Language Processing. As Visual Prolog also implemented some of the object oriented programming features, it also rectifies some of the Prolog's limitations.

Limitation

Visual Prolog is harder to be understood as a result of the merge between Prolog language and object oriented methodology. Granted, it does address both methods' weaknesses. However, it also introduces more limitations as it does so.

2.4.3. Methodology Review

Comparison between Prototyping and System Development Life Cycle (SDLC).

Complaints about going through the SDLC centres around two main concerns, which are interrelated. The first concern is the extended time required going through the development life cycle. As the investment of the analyst time increases, the cost of the delivered system rises proportionally.

The second concern about using SDLC is that user requirements change over time. During the long interval between the time the user requirements are analysed and the time that the finished system is delivered, user requirements are evolving. Thus, because of the extended development cycle, the resulting system maybe criticised for inadequately addressing current user information requirements.

It is apparent that the concerns are interrelated, since they both pivot on the time required to complete the SDLC and the problem of falling out of touch with user requirements during subsequent development phases. If a system is developed in isolation from users (after initial requirements analysis is complete), it will not be up to their expectations.

A consequence of the problem of keeping up with user requirements is the suggestion that users cannot really know what they do or do not want until they see something tangible. And in this traditional SDLC, it is often too late to change an unwanted system once it is delivered.

To overcome these problems, some analysts propose that prototyping be used as an alternative to the system development life cycle. When prototyping is used in this way, the analyst effectively shortens the time between ascertainment of information requirements and delivery of a workable system. Additionally, using prototyping

instead of the traditional system development life cycle might overcome some of the problems of accurately identifying user information requirements. *(Kendall & Kendall, 1999)*

2.4.4. Existing Natural Language Processing System

Start Natural Language Question Answering System
http://www.ai.mit.edu/projects/infolab/start-system.html

The START server answers questions in English about the MIT AI Laboratory, geography, and assorted other topics. The Server is based on the START natural language system developed by Boris Katz and his associates in the Infolab Group START has been available to World Wide Web users since December 1993.

Brainhat Natural Language Processing-based Operating System
http://www.brainhat.com

Brainhat is an NLP executive that can dispatch tasks, handle natural language events, evaluate ideas, and ask and answer questions. Because Brainhat is knowledge representation-based, it can digress within a conversation, answer questions outside the script, and keep rich context.

Geobase
http://www.pdc.dk/

Being the developer of the Visual Prolog, the Prolog Development Centre have provided a sample of a NLP interface to a geographical database developed using Visual Prolog.

2.4.5. Interview with users

There were a few interview sessions done with a few users to get a clearer idea of what is expected from the project. From what I understand, there are a few important aspects, which needed specific consideration. The areas are highlighted below :-

attractiveness

One of the key factor in designing web sites is the attractiveness of the Graphical User Interface (GUI). If the users find screens appealing, they are likely to be more productive. Screens should draw users into them and hold their attention. A screen should never be crowded. Using multiple screen windows or hyperlinks is far better off then jamming everything onto a screen or page. By creating screens that are easy to grasp at first glance, one appeals to both inexperienced and experienced users. With the advent of Graphical User Interface (GUI) it is possible to make input screens very attractive.

Examples:

Different Types Fonts

Types of fonts are another way to make screens attractive to users. Different styles enhance differentiation among categories. For instance thick, sans serif type styles can be used to denote main categories and to give screens a modern look. Larger type can indicate captions for data entry fields. Thinner type with serifs can be used to designate subcategories on the same screen and provide a more conservative look. *(Kendall & Kendall, 1999)*

Using colour in Screen Design

Colour is an appealing and proven way to facilitate computer input. Appropriate use of colours on display screens allows you to contrast foreground and background, to highlight important fields, to feature errors and to call attention to many other special attributes.

Highly contrasting colours should be used for foreground and background. This helps users grasp what is presented quickly without straining. Specifically the top five most legible combinations of foreground lettering on background are (starting with the most legible combination):

- *Black on yellow*
- *Green on white*
- *Blue on white*
- *White on blue*
- *Yellow on black*

As can be gathered from these possible foreground and background combinations, bright colours should be used for foregrounds, with less bright colour for the background.

Colours should be used to highlight important fields on screens. Fields that are important can be coloured differently from the rest. When creating web-based applications, hyperlinks are usually colour-coded to show users that a hypertext path can be taken. Hyperlinks can be made to change colour after a user has clicked on them once with the mouse. This colour-coding prevents users from pursuing previously used hyperlinks and thus helps organise their search and save them valuable time.

User friendly

The system created should not involve any complications when using it. Users of the system can involve inexperienced or experienced users. Therefore, the system should be made simple and precise.

2.4.6. Programming Tools

Visual C++ 6.0

Microsoft was the company that developed Windows. After going through a successful version up till version 2.0, Microsoft realised that in order to provide more Windows based functionality to the developer, more coding have to be done. This is quite troublesome considering that some of the functions required thousands lines of code to just display something like a box.

Thus, to solve this problem, in conjunction with its Windows 3.0 release, Microsoft also released a development tool called Visual C++. By releasing a development tool, it has ensured a future for the release of the subsequent Windows versions.

In 1995, Microsoft also released its Microsoft Foundation Class (MFC) in which all of Microsoft development tools support. The class is also supported by other software development vendors, such Borland(formerly Inprise) and Intinuit.

The MFC enable the programmer to easily call Windows functions by stating only a few commands, leaving the codes used to implement them to the operating system to execute.

CHAPTER 3
METHODOLOGY

3.1. Project Description

As mentioned earlier report, E. Help Desk consists of two important modules, which are the public user interface module, the administrator module.

The public user module consists of a main feature, which are the search facility

The administrator module on the other hand, consists of a data information keying system.

In conclusion, E. Help Desk is a basic search engine using Natural Language Processing.

3.2. Approach to System Development

The approach used in developing E. Help Deskis prototyping. Generally, there are four guidelines in developing a prototype.

- **Work in manageable modules**
- **Build the prototype rapidly**
- **Modify the prototype in successive iterations**
- **Stress the user interface**

Working in manageable modules

A manageable module is one that allows users to interact with its key features yet it can be built separately from the other system modules. Module features that are deemed less important are purposely left out of the initial prototype.

There were two public user interface module, the dealer module and the administrator module. Since each of these modules is quite wide, they were divided into a few sub-modules which were developed separately.

Although on the surface, it seems that the modules only employ few methods, but in the implementation, the sub modules were necessary as to facilitate problem solving. This is in line with the Object Oriented Programming approach.

Building the Prototype Rapidly

Speed is essential to the successful prototyping of an information system. After a brief analysis of information requirements, working models for the prototype were constructed. The prototype took less than a week to put together. Putting together an operational prototype both rapidly and early in the system development life cycle allows the analyst to gain valuable insight into how the following steps in the project should go.

Time allocation for building each sub-module should be short since there are many other sub-modules need to be designed. To manage time, a Gantt chart was used. With software like Visual C++ 6.0, a prototype can be developed quickly as it is easy to learn and use.

Modifying the Prototype

A third guideline for developing the prototype is that its construction must be able to support modifications. Making the prototype modifiable means creating it in modules that are not highly interdependent. If this guideline is observed, less resistance is encountered when modifications in the prototype are necessary.

Once each of the modules were developed, they were tested with users so that they meet the user requirements. If the users were not satisfied with certain criteria, the modules were then modified according to their needs. This process took place for each module before they were linked together so that they will not be highly interdependent.

The prototype is generally modified several times, going through several modifications. Change in the prototype should move the system closer to what users say is important. Each modification necessitates another evaluation by users.

The user's interface with the prototype and eventually the system is very important. At this stage the goal of the analyst is to design an interface that both allows the user to interact with the system with a minimum of training and fuss; and allows a maximum of user control over represented functions.

3.2.1 Prototyping Stages

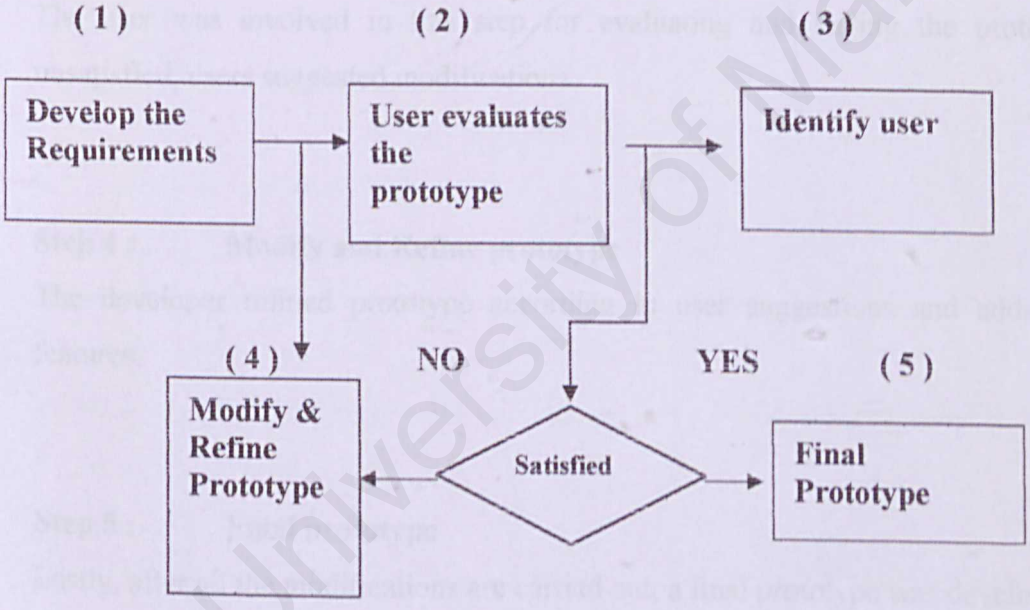

Figure 3-1 : Stages In Prototyping

Step 1 : Identify User Requirements

Beginning of the development, whereby, gathering of requirements was done to come up with a requirement specification stating all requirements of users from the system.

Step 2 : Develop the prototype

A prototype system was constructed.

Step 3 : User evaluates the prototype

The user was involved in this step for evaluating and testing the prototype. If unsatisfied, users suggested modifications.

Step 4 : Modify and Refine prototype

The developer refined prototype according to user suggestions and adds in new features.

Step 5 : Final prototype

Lastly, after all the modifications are carried out, a final prototype was developed.

3.3. Justification

3.3.1 Advantages of Prototyping

- system developed more quickly
- system respond to changes more easily
- users are more involved in the system ,so the developer spends less effort in information gathering
- many end-user computing tools and software are suitable to be used with prototyping
- development costs are low

3.3.2 Application/suitability

- simple systems
- small systems
- when the user is not able to state all the requirements in the beginning
- for risky systems

3.4. System Requirements

It is necessary to draw out the system's requirements to provide a guideline for developing the system. A requirement is a feature of the system or description of something the system is capable of doing in order to fulfill the system's purpose.

The system's requirements for this project were determined for the three separate modules. The type of requirements for a project is normally separated into functional requirements and the non-functional requirements.

3.4.1 Functional Requirements

A functional requirement describes an interaction between the system and its environment. It also describes how the system should behave given a certain stimuli. The functional requirements for E. Help Desk were separated between the two modules.

- **A Public User Module**

This module accepts questions from the user and displays the appropriate answer. This module is divided into three sub modules

a) Question Field - The users of the systems inputted their questions here.
b) Answer module - This module display the appropriate answers to the questions.
c) Linking Module - In this module, the questions are broken into string tokens which is then searched in the database. This module also format the answers according to the type of question asked.

- **Administrator Module**

In this module, only the information needed to be updated into the user module are entered.

3.4.2 Non-Functional Requirements

A non-functional requirement describes a restriction on the system that limits the choice for constructing a solution to the problem. These solutions will narrow down the selection of programming languages, platform or implementation techniques or tools.

Among the non-functional requirements of E. Help Deskare :-

Graphical User Interface (GUI)

Graphical User Interface allows direct manipulation of the graphical representation on the screen, which can be accomplished with keyboard input or a mouse. The key to GUI is the constant feedback on task accomplishment that it provides. The users of Web sites are unknown to the developer, so a design must be clear-cut.

User Friendly

Associated to the previous requirement, this function also allows users to operate the system with ease. This is done by providing the necessary commands, help and display feedback

3.5 Tools Used

3.5.1 Hardware

1. A computer with a minimum requirement of hardware as below was needed to develop E. Help Desk:

- 32 MB of memory
- 600 MB of storage
- a Pentium 200

3.5.2 Software

- ***Microsoft Visual C++ 6.0***

For Client applications (Interface and Processing)

- ***Microsoft Internet Explorer version 5.0***

As a Browser to view documents saved from the internet

- ***Adobe Acrobat Reader***

Used to read other documentations from the internet

- ***Microsoft Windows Millennium***

Development Operating System

- ***Microsoft Windows 98***

Other Operating System Where The System Was Also Tested

- ***Microsoft Word 2000***

Documentation

Why Microsoft Visual C++ 6.0 was used

Visual C++ 6.0 was chosen as the as the main tool for the development of **E. Help Desk** because of it's the most used software platform used to develop Windows Application.

3.5.3 Programming Tools

- ***Visual C++ 6.0***
- ***Visual Basic 6.0***

CHAPTER 4
SYSTEM DESIGN

Design phase is the stage of system development where the requirements for the system are translated into the system characteristics.

4.1 System Design

4.1.1 Prerequisites for Systems Design

System design is conducted after careful evaluation of the following requirements:

- ◊ ***User requirements***
- ◊ ***Hardware requirements***
- ◊ ***Systems requirements***

User Requirements

In designing new systems, the analyst must consider the requirements of the major user, as well as other users, and determine the extent of their dependence on the new system. Cost is often a major constraint, which implies that a new system seldom meets the requirements of all users. In the end, a balance between cost and performance must be incorporated for an overall satisfactory system design.

In determining the user's requirements, an understanding must be reached as to what can be expected of the system. Although it is the users responsibility to spell out what is needed, this phase is often delegated to the system's analyst for a final decision. Thus, the computer professional whose orientation is of analysis and design of systems conveniently specifies the user's requirements.

Hardware Requirements

Systems design requires an evaluation of the processing methods used by the new system. Of the methods available, computer-based systems design is the most complex. Then, it is important that the systems analyst be familiar with the computer system's capabilities and limitations and alternative ways of achieving optimum efficiency of system operation.

Systems Requirements

The primary systems requirements are economy, flexibility, reliability, simplicity, and acceptability. That is a system must be economical to operate, flexible enough to accommodate future change and capable of producing reliable output acceptable to the user.

Economy

Traditionally, the analyst is expected to design a system that will meet the user's requirements at the lowest cost possible.

Flexibility

The new systems should incorporate features that make it possible to modify any aspect of the system when necessary. In addition to meeting the user's present requirements, new systems should accommodate changes in future requirements without the need for a major or frequent update.

Simplicity

A simple, yet flexible system designed to serve the needs of the user at a reasonable cost would be ideal.

Reliability

Reliability refers to the confidence the user has in the output of the new system. An analyst must assume a total responsibility for securing a reliable system.

Acceptability

Throughout the design phase, it is extremely important to secure the acceptance and support of all users who will be affected by the new system. The performance of most reliable systems can also be threatened without the user's cooperation and support.

To summarize, a new system is designed through a creative effort of the analyst. A new system should satisfy a particular user's requirement without adversely affecting the information's requirement of the users.

4.1.2 Architectural Design

The primary objective of architectural design stage is to develop a modular program structure that represents the control relationships between modules. In addition, architectural design melds program structure and data structure defining interface that enables data flow throughout the program. Diagrams below represents the architectural design for E. Help Desk, public user module, the dealer module and the administrator module.

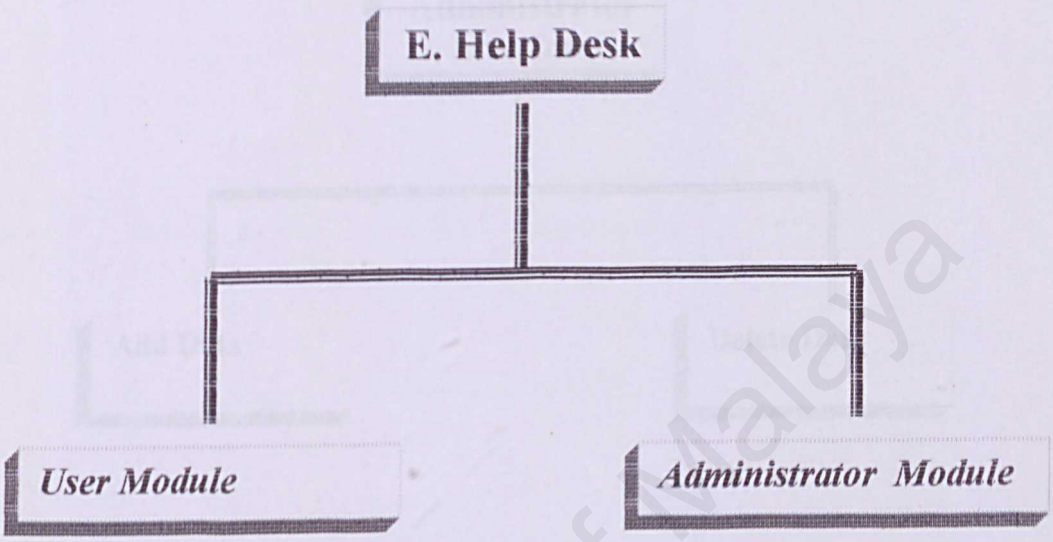

Figure 4-1: Architectural Design of E. Help Desk

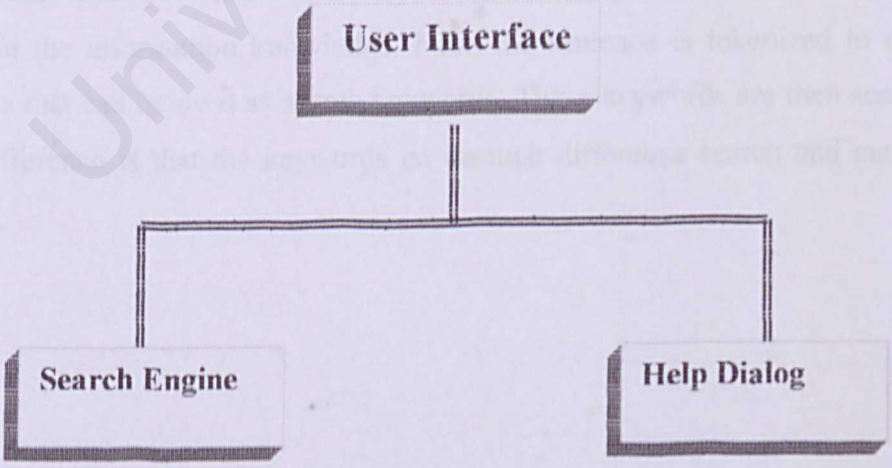

Figure 4-2 : Architectural Design of User Module

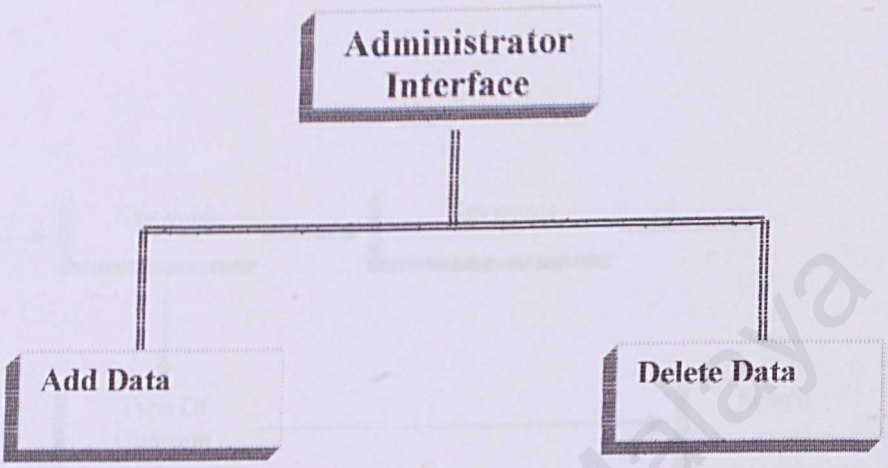

Figure 4-3 : Architectural Design of Administrator Module

4.2 Search Engine Strategy

E. Help Desk uses keywords searching to shift through the information that is contained in the information knowledge. First, the sentence is tokenized to extract word tokens that can be used as search keywords. These keywords are then searched; the only difference is that the keywords go through difference search and matching procedures.

The procedures can be shown by the diagram below:

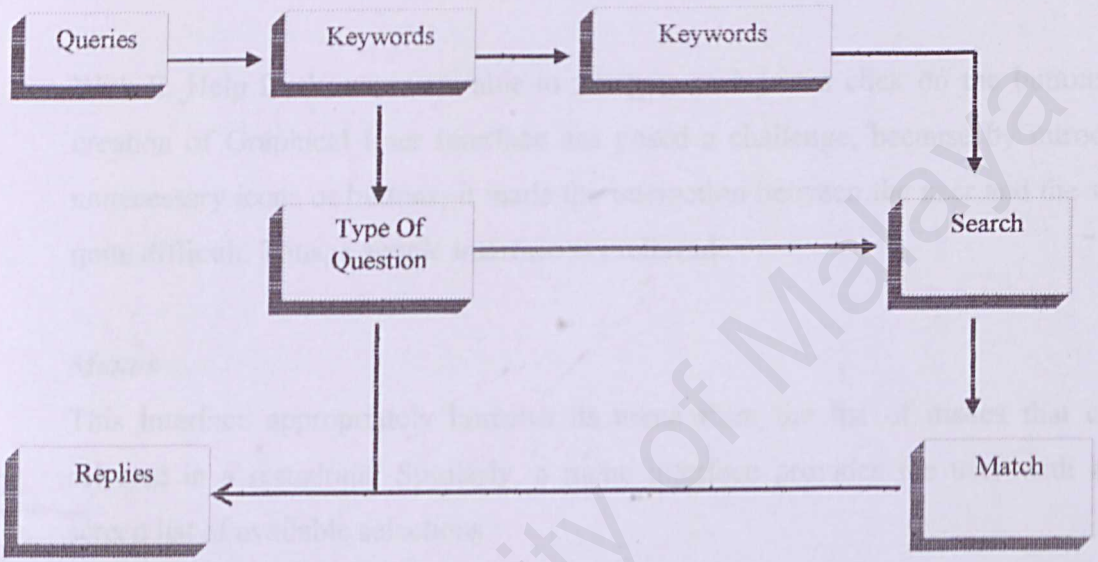

Figure 4-3 : Architectural Design of Search function

4.3. User Interface Design

4.3.1. Types of User Interface

There are several different kinds of user interface are considered when preparing for the interface for E. Help Desk. Even though Graphical User Interface (GUI) is the main interface used for E. Help Desk, but for the whole system, a selection of interfaces is applied. A list of interfaces is listed below.

Graphical User Interface

Graphical User Interface allows direct manipulation of the graphical representation on the screen, which can be accomplished with keyboard input, or a mouse. Direct manipulation requires more system sophistication than other user interfaces.

With E. Help Desk, users are able to navigate with just a click on the buttons. The creation of Graphical User Interface has posed a challenge, because by introducing unnecessary icons or buttons, it made the interaction between the user and the system quite difficult. Thus, a simple interface is preferred.

Menus

This interface appropriately borrows its name from the list of dishes that can be selected in a restaurant. Similarly, a menu interface provides the user with an on-screen list of available selections.

In responding to the menu, a user is limited to the options displayed. The user need not to know the system but does need to know the task should be accomplished. To best utilise the menu, users must know which task they desire to perform.

In administrator mode, there a few important forms are prepared for the administrator to change the contents of the database. Most of these text fields have brief explanation to facilitate the administrator. This is done to save time and avoid errors. E. Help Desk consists of only very simple forms for users and administrator, therefore there will not be any complications involved in submitting the correct and viable details.

4.3.2. Dialog and Desktops

Dialog is the communication between the computer and a person. Well-designed dialog makes it easier for people to use a computer and leads to less frustration with the computer system. There are several key points for designing good dialog. They include:-

1. Meaningful communication, so that user understands what people are entering and people understands what the computer is presenting or requesting.
2. Minimal user interaction
3. Standard operation and consistency

Communication

In E. Help Desk, information is presented clearly to the user. This includes presenting an appropriate title on each screen, minimizing the use of abbreviations, and providing a clear user feedback.

User instructions (most of them are in one line) are supplied regarding details on each screen of E. Help Desk.

Minimal User Action

Keying is often the slowest part of a computer system, and fast and easy interface will minimize the time needed to complete and inquiry. In E. Help Desk, this is accomplished using many different ways.

1. Keying few words instead of whole sentence.

In E. Help Desk, other than keying in the whole sentence, users can use just type in the keywords that will be used for searching the knowledge base. Words such as 'is' and 'are' can be safely omitted.

2. Retrieving data that is already stored.

For E. Help Desk administrators, there is not much work involved in deleting and modifying information. Thus, skeletal information on the text fields are provided. All the relevant information on the cars will be displayed in a form. Then administrator can just delete or modify necessary information.

Standard Operation and Consistency

The operation should be consistent throughout its set of different screens and in the mechanisms for controlling the operation of the screen throughout different applications. Consistency makes it easier for the users to learn how to use new portions of the system once they are familiar with one component. Consistency is achieved in E. Help Desk

CHAPTER 5
SYSTEM DEVELOPMENT AND TESTING

Usually, in a software prototyping project, the requirement analysis, system design and development phases do not have a clear boundary. Each phase tends to involve one another. System development is a process that while converting the system requirements and designs into program codes involves some modifications to the previous design.

System development translates to a detail representation of the software into a programming language realisation. The translation process continues when a computer accepts a source code as an input and produces a machine code. In order to carry that out, appropriate tools and suitable languages are needed to code the programs. A number of software tools were chosen in the development of E. Help Desk.

5.1 Development Environment

5.1.1 Software Tools Requirements

1. Software Tools for Design and Report Writing

The design process involves the drawing of structure charts, data flow diagrams and other drawings that form the foundation of the software development. The purpose of this graphic based logical design is to provide an overall view of the system and interconnection between the modules. The tools used here are Microsoft PowerPoint 2000 and Microsoft Word 2000 for Windows.

2. Software Tools for Development

During the course of development for E. Help Desk, a vast array of software tools were used. Table 5.1 below depicts the software used to develop the system.

Table 5.1: Summary of Software Used

Software	Module	Description
Microsoft Windows 98	System requirement	Operating system used for testing compatibility
Microsoft Windows Millennium	System requirement	Operating system used for development
Microsoft Visual C++ 6.0	System requirement	Development tool

5.2 Development of E. Help Desk

5.2.1 Application Development

As much of the system's development have been covered earlier, the authour considered it suffice to say that it was done using the appropriate methods available.

5.2.2. Coding Approach

Good programming skills produce a reliable and easy to maintain system. A good coding style always requires :-

1. Readability

The source code should be able to be read by other programmers and also non-programmers without any difficulties. This requires :-

- Selection of identifier (variables and labels name)
- Composition of comments
- Organization of the overall program

2. Good Naming Technique

This means that names given to variables, controls and modules should provide for easy identification for the programmer. The naming convention should be created with coding consistency and standardization in mind.

3. Internal Documentation

Internal documentation contains information directed at someone who will be reading the source code of your programs. Thus, summary information is provided to identify the program and describe its data structures, algorithms and control flow. Usually, this information is placed at the beginning of each component in a set of comments called the header comment block.

An example of header comment block is attached in the appendix. It is written in the sample source code for random function.

Coding in E. Help Desk provides an internal documentation so that other programmers easily understand the codes.

4. Modularity

In a modular design, the components have clearly defined inputs and outputs, and each component has a clearly stated purpose. Thus, it is easy to examine each component separately from the other to determine whether the component implements its required tasks. Moreover, modular components are organized in a hierarchy, as a result of decomposition or abstraction, so that we can investigate the system one level at a time. For these reasons, we try to design our software so that it is as modular as possible.

Components are said to be arranged in different levels of abstraction. The levels of abstraction help us to understand the problem addressed by the system and the solution proposed by the design. By examining the levels from the top and working down, the more abstract problems can be handled first and their solution carried through as the detailed description is generated.

Modularity also hides details. An advantage of this information hiding is that each component hides a design decision from others. Thus, if design decisions are likely to change, the design as a whole can remain intact while only the component design changes.

Abstraction and information hiding allows us to examine the ways in which components are related to one another in overall design. We strive in most designs to make the components independent of one another.

To recognize and measure the degree of component independence in a design, we use two concepts: coupling and cohesion.

Coupling

We say that two components are highly coupled when there is a great deal of dependence between them. Loosely coupled components have some dependence, but

the interconnections among components are weak. Uncoupled components have no interconnections at all; they are completely independent.

Cohesion

In contrast to measuring the interdependence of components, cohesion refers to internal "glue" with which a component is constructed. The more cohesive a component, the more related the internal parts of the component are to each other and to its overall purpose. In other words, a component is cohesive if all elements of the component are directed toward and are essential for performing the same task.

5.2.3 Coding Style

Coding style is an important attribute of source code and it determines the intelligibility of a program. An easy to read source code makes the system easier to be maintained and enhanced. The elements of style include internal (source code level) documentation, method for data declaration and approach to statement construction.

The following lists down some of the styles used during the coding of E. Help Desk:-

1. Selection of meaningful identifiers (variables and labels) name.
2. Description and appropriate comments, written in the source code.
3. Indentation of codes increases the readability of source code.

5.3 System Testing

Testing is performed to ensure that the programs are executed correctly and conforms to the requirements specified. It provides a method to uncover logic error and for

testing system reliability. The strategies used for testing are unit testing, integration testing and system testing.

Validation

System testing validates the requirements. Validation ensures that the system has implemented all of the requirements, so that each system function can be traced back to a particular requirement in the specification. That is, validation makes sure that the developer is building the right product (according to the specification).

Verification

System testing also verifies the requirements. Verification ensures that each function works correctly. Verification checks the quality of the implementation.

For E. Help Desk, in general 3 types of testing are performed:-

- Unit Testing
- Integration Testing
- System Testing

5.3.1 Unit Testing

Steps the authour has taken in unit testing:-

1. First, the code is examined by reading through it, trying to spot algorithm, data, and syntax faults. Comparing the code with the specifications and with the design was also be done to make sure that all relevant cases are considered.

2. Next, the code is compiled and remaining syntax faults are eliminated.

3. Finally, test cases are simulated to show that the input is properly converted to the desired output is developed. This pertains to testing the system by using it.

In unit testing, each of these steps one at a time was examined.

During the unit testing phase of E. Help Desk, all small or big functions and subroutines were tested to check for coding or logical errors. The three basic steps stated above were followed during the unit testing of E. Help Desk. Often, the logical errors were the most difficult to rectify as it's quite hard to detect. Often, the logical error will cause the application to crash during the execution.

5.3.2 Integration Testing

In E. Help Desk, this strategy involves combining modules one by one, that is using the incremental integration approach. Therefore, the system that is constructed is tested in small segments where errors are more likely to be tested completely because they are easily isolated and corrected and a systematic test approach may be applied. Moreover the interfaces are more likely to be tested completely too.

5.3.3 System Testing

System testing is actually a series of different test whose primary purpose is fully exercising the computer-based system. System testing is to ensure that the system is

functioning well under a larger system. Performing system testing on E. Help Desk was to ensure that all system elements have been integrated and perform the functions as required.

System Testing Process

There are many steps in testing a system :

1. Function testing
2. Performance testing
3. Acceptance testing
4. Installation testing

Performance test of E. Help Deskcompares the integrated components with the nonfunctional system requirements, including accuracy, speed and reliability.

An acceptance test was done to make sure that the system meets users understanding of the requirements, which may be different from the users. For E. Help Desk, some of my friends were asked to test the system, so that they can give their opinion regarding the system.

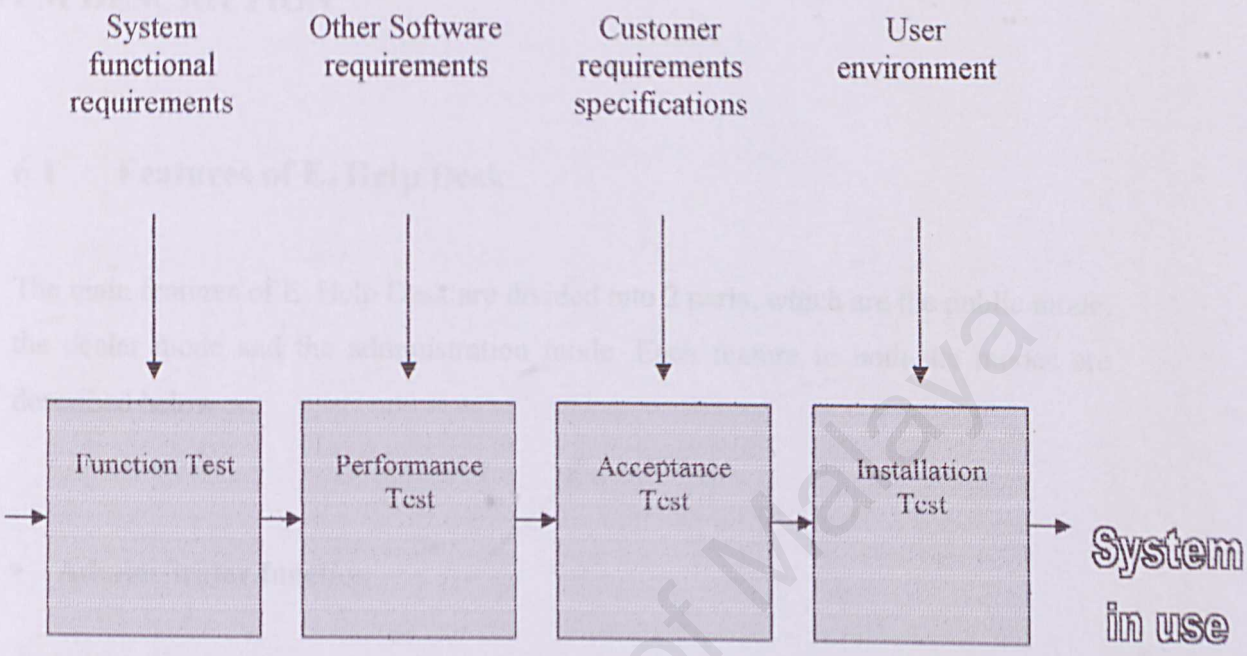

FIGURE 5.1 : Steps in testing process

CHAPTER 6
SYSTEM DESCRIPTION

6.1 Features of E. Help Desk

The main features of E. Help Desk are divided into 2 parts, which are the public mode, the dealer mode and the administration mode. Each feature in both the modes are described below :-

- **Administrator function**

Update data

Whenever there is a need to update existing data, administrator can select any particular data and modify it. Besides that an administrator can also update its specifications and characteristics by filling in the appropriate fields in the form.

- **Public User Functions**

Search

A user is able to search for his or her particular query by keying in particular keywords in the search text box and then clicking on the answer button. The system will soon provide the fill list of details for the closest match as requested by the user. A user can either key in using uppercase or lowercase letters. If the cars typed in are not in the database, the system will notify the user about it.

6.2 System Strengths

1. **Speed of rapid development.**

   E. Help Desk was developed in about three months. In real life, speed rapid development is important because systems that usually require a gestation period to develop may be outdated even before it is ready to be used.

2. **Windows Platform**

   Being developed for Windows based computer, E. Help Desk enjoys the ease of use as Windows platform is the most popular operating system used by most people (about 80% of PC users).

3. **User Friendly Interface**

   E. Help Desk has a very user-friendly interface. It is easy to use even by inexperienced users. All the features are labeled in the simplest way possible, so that users will not be left wondering about the use of each feature. With interesting icons and titles describing each feature, users will find E. Help Desk an interesting search system to use.

4. **Ease of control and manipulation.**

   Since E. Help Desk uses a simple approach, it is easy to use and control each feature. Users can easily choose an appropriate way to search.

5. **Case Insensitive.**

   Users are allowed to search with lowercase letters, uppercase letters or a mix of cases. The database will be able to find the matches, regardless of the case typed in.

6. **Help Dialog**

E. Help Desk also has a help dialog feature, which can help users if they need any help while using E. Help Desk. This is provided so that E. Help Desk can be of full use to the users.

7. Maintenance

E. Help Desk has an administrator menu, which allows administrator to add new data, update existing data and delete any unwanted data. All these can be done without accessing the database because all these changes are allowed in the interface. Administrator need not worry to perform these entire tasks since they can be done in a very simple way. For example, to delete and modify any data, the particular data can be selected on the interface itself.

6.3 System Limitations

1. *Lack of other interactions*

The system is supposed to also provide other interactions, such as displaying the location map of the area or receiving the question via voice recognition and speech synthesis. The system's ideal implementation will be in a networked environment where it acts as a search engine for the user to ask questions which will be answered specifically by type of question, in such cases displaying multi media or database capability if possible.

2. *Limited Graphic Illustrations*

Sometimes, users will understand the system better if graphical illustrations or even photos of the description were provided.

CHAPTER 7
CONCLUSION

7.1 Problems encountered

1. Lack of knowledge in applications development.

Source Coding.

The biggest problem that the authour encountered throughout this coursework is the lack of good reference books from the library and other sources such as the internet. What was meant by this is that although the books themselves provide good instructions, often they do not describe how to adapt the examples in real world applications. Many of the examples are mostly for classroom learning, which only give the reader a basic understanding of the development systems. The examples for other sources were also plagued by this problem.

7.2 Solutions to problems

1. Books

   To learn how to code in Visual C++, books on C, C++ and Visual C++ were used. This helped me to understand the Visual C++ commands.

2. Searching on the internet

   Some of the web sites referred helped the authour to gain knowledge on algorithms that may be used on my system. Programming design sites helped me on the design aspect of E. Help Desk.

3. Discussions with supervisor

   Through discussions with the authour's supervisor, the authour has learnt the best way to present the information to users.

7.3 Knowledge Gained

Throughout the entire development phase of E. Help Desk, there was no doubt a lot of knowledge was gained. Among them are:

1. Learnt additional software tools.

Languages like Visual C++ were used to create web application.

2. Mastered the process of creating software.

The whole process of creating software was implemented, beginning from the user's requirements to the analysis, system design, coding and finally the testing of the software.

3. Skills in gathering information and facts.

Information needed for E. Help Desk was collected not only from one source, but also from several places and people. This therefore improved my communication skills and interpersonal skills.

4. Experience in problem solving

Problem solving here refers to the implementation of a system, which is the encoding of it.

5. Learn to work independently.

This experience was very different from other assignments, as I had to work independently and all sorts of new tensions were also experienced.

6. Skills in writing documentation.

I gained knowledge of the proper format of documentation.

7.4 Conclusion

This system was completed successfully with its strengths and limitations as mentioned earlier. Generally, it is a user-friendly system and is easily understood. Through this system, the fact that the software engineering part of a system was understood and this will help a lot in developing future projects.

APPENDIX A

SCHEDULE

Figure A-1 below shows the schedule for the development of system

30th May 2001 - 30th August 2001

Identify User Requirements

During this period of time , the focus was mainly on identifying user requirements. At the beginning of this period , the objectives, scope and significance of were specified so that the authour knew what he's going to do. Once this was done,a review of literature was carried out. Review of Literature was followed by review on methodology. In this phase, project description, approach, justification, development strategy, proposed tools and the statement of expected outcome were specified.

30th August 2001

Submission of Proposal

29th October 2001 – 20th November 2001

System design and development of prototypes.

During this period of time, all the prototypes were developed starting with the public user module followed by administartor module.

20th November 2001 - 27th December 2001

User Evaluates The Prototype

Identify what exactly the users want and if the users are not satisfied , prototypes are modified.

27th December 2001 – 4th January 2001
Prototypes modified and refined

After listing out the flaws in the prototypes developed , they are soon modified according to the users need and specifications.

1th January 2002 – 20th January 2002
Development of final prototypes and system testing

If the users are satisfied with the refined prototype , a final prototype will be developed and all the prototypes will be linked together. Prototypes are tested before and after they are linked . Lastly, system testing was carried to confirm that it is error free.

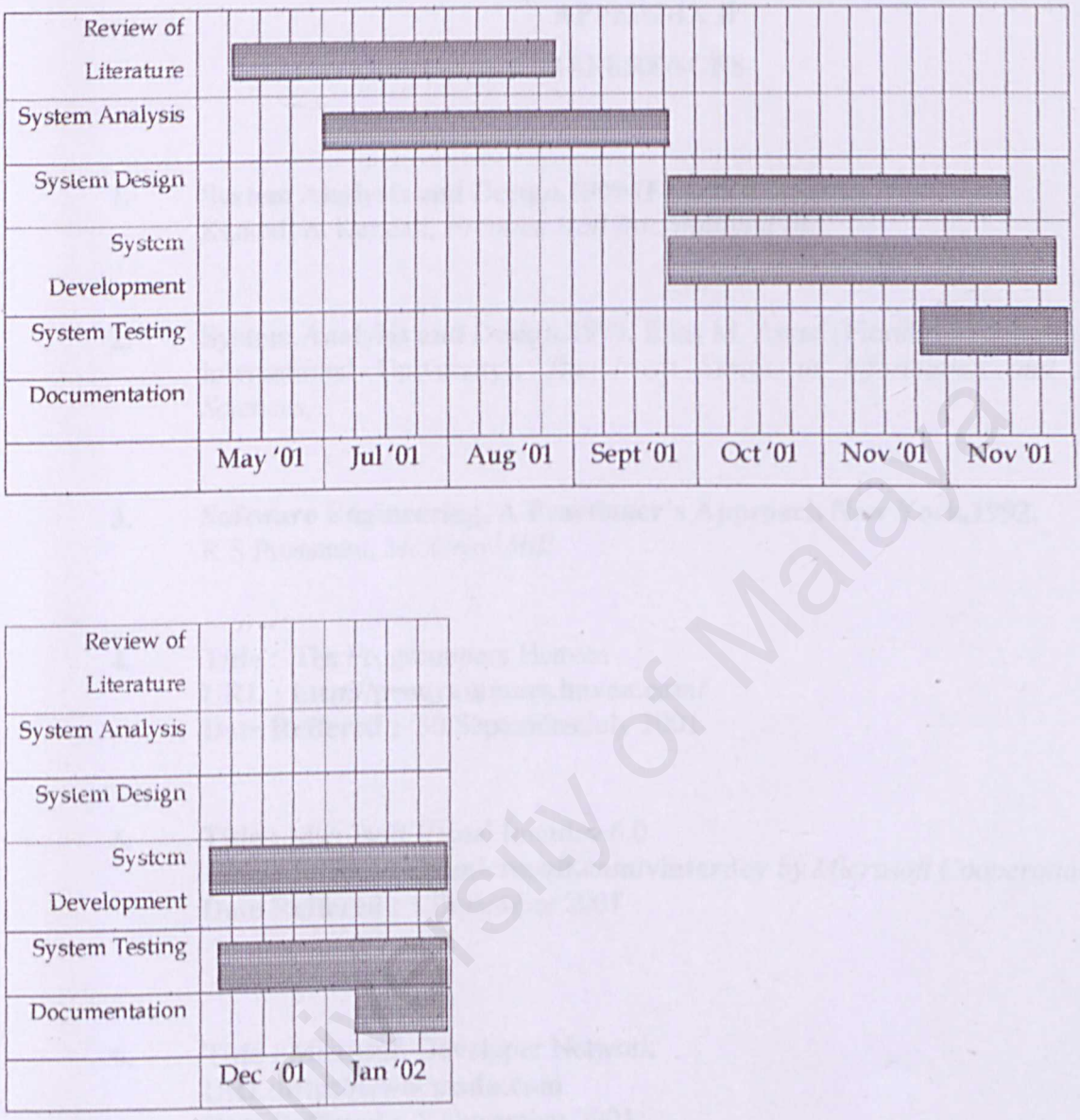

Figure A-1 Project Schedule

APPENDIX B

REFERENCES

1. **System Analysis and Design.1999 (Fourth Edition),** Kendall & Kendall, *Prentice Hall International Inc.*

2. **System Analysis and Design.1979,** Elias M. Awad (Florida International University), *The Irwin Series in Information and Decision Sciences.*

3. **Software Engineering, A Practioner's Approach New York,1992,** R.S Pressman, *Mc.Graw Hill.*

4. **Title :** The Programmers Heaven
   URL : http://programmers.haven.com/
   Date Reffered : 30 SeptemberJuly 2001

5. **Title :** Microsoft Visual Interdev 6.0
   URL : http://msdn.microsoft.com/vinterdev *by Microsoft Cooperation*
   Date Reffered : 1 December 2001

6. **Title :** Microsoft Developer Network
   URL:http://www.msdn.com
   Date Reffered : 8 November 2001

7. **Title :** Brainhat
   URL: http://www.Brainhat.com
   Date Reffered: 12 August 2000

8. **Title : Prolog Development Centre**
   URL: http://www.pdc.dk
   Date Reffered : 12 August 2001

9. **Title:**CNET
URL:http://www.cnet.com/
Date Reffered : 10 October 2001

10. **Title:**Annette Blanar Steadle Technical Writing Consultant
URL:http://members.aol.com/absteadle
Date Reffered : 10 October 2001

Introduction

E. Help Desk is a Natural Language based-search engine for answering normal from the user.. It has two main modules which is the user module and the administrator module. The user module caters to question from the users whereas the administration modules provide means for updating new information inside the knowledge base. This User Manual contains a step-by-step guide for beginners on how to it.

System Installation

The system can be copied using normal copy or installation wizard.

Hardware Requirements

The hardware configuration requirements for accessing E. Help Desk are as follows:

- Pentium 166 MHz processor
- 16MB RAM
- 14" Colour SVGA Monitor (with resolution set to 800 x 600) - 17' inch recommended
- Mouse (for administrator module, recommended for the user module)

- Windows keyboard

Software Requirements

E. Help Desk requires the following software to access its web page:

- Microsoft Windows 95 (or later)

How to access the E. Help Desk User module

- Run the user module's exe file

How to access the E. Help Desk Administrator module

- Run the administrator module's exe file

How To Use The User Module

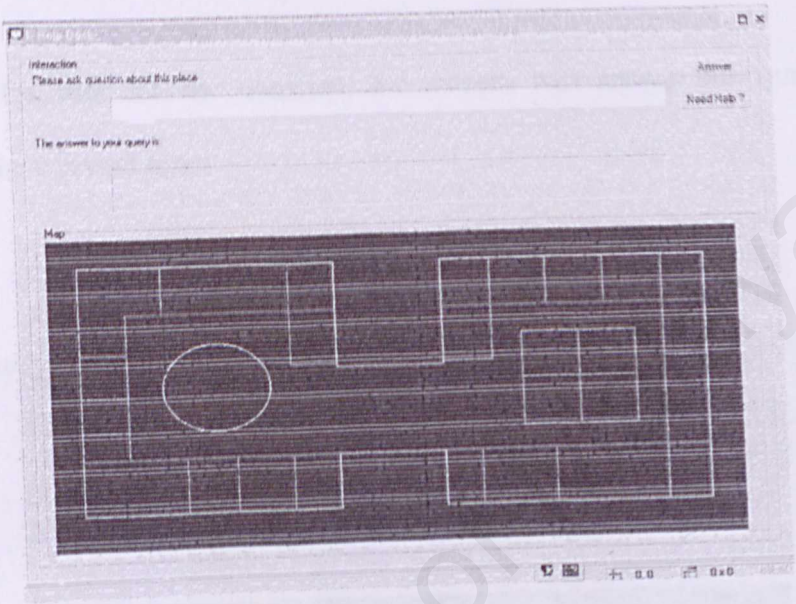

Figure 1 : The module for the user

As shown in the first figure, this is the user module. Here, a map of the building is shown as the intended application is for answering question based on the building's tenant. Above the map there are two text fields. The first one is the query field where user enters the question and another one is the answer field where the answer will be displayed.

Beside the two fields are two buttons which is ask and help. The ask button is for asking the system to return the query's question where as the help button will bring about the help dialog.

Notice that the interface will be kept to a minimum as the user will not be expected to learn how to use the system . Instead, it is is designed to be more intuitive.

After the user ask the question, the system will answer the question, while highlighting the relevant place

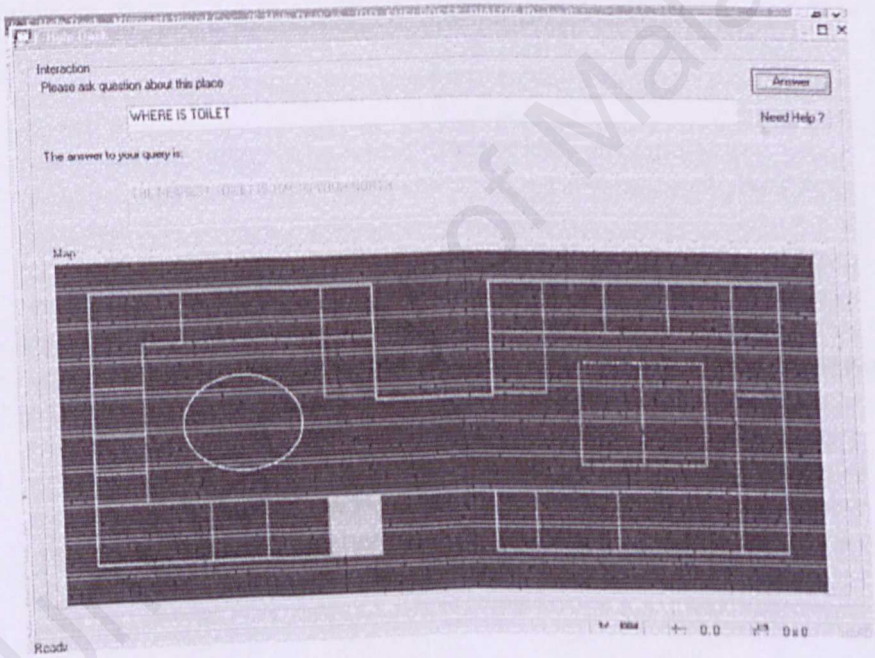

Figure 2 : Example of the module in action

Notice that the place is highlighted where the probable place is situated.

How To Use The Administrator Module

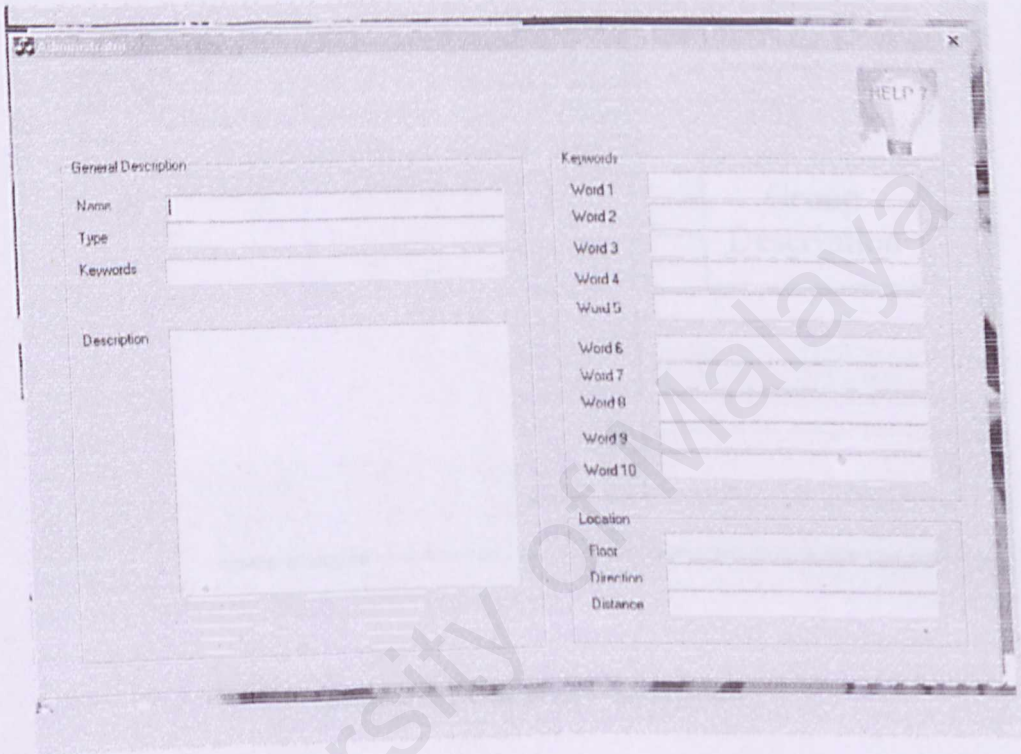

Figure 3 : The administrator's module

This is the administrator's module where the information for the knowledge base is updated. Notice that there are three information groups and a help button.

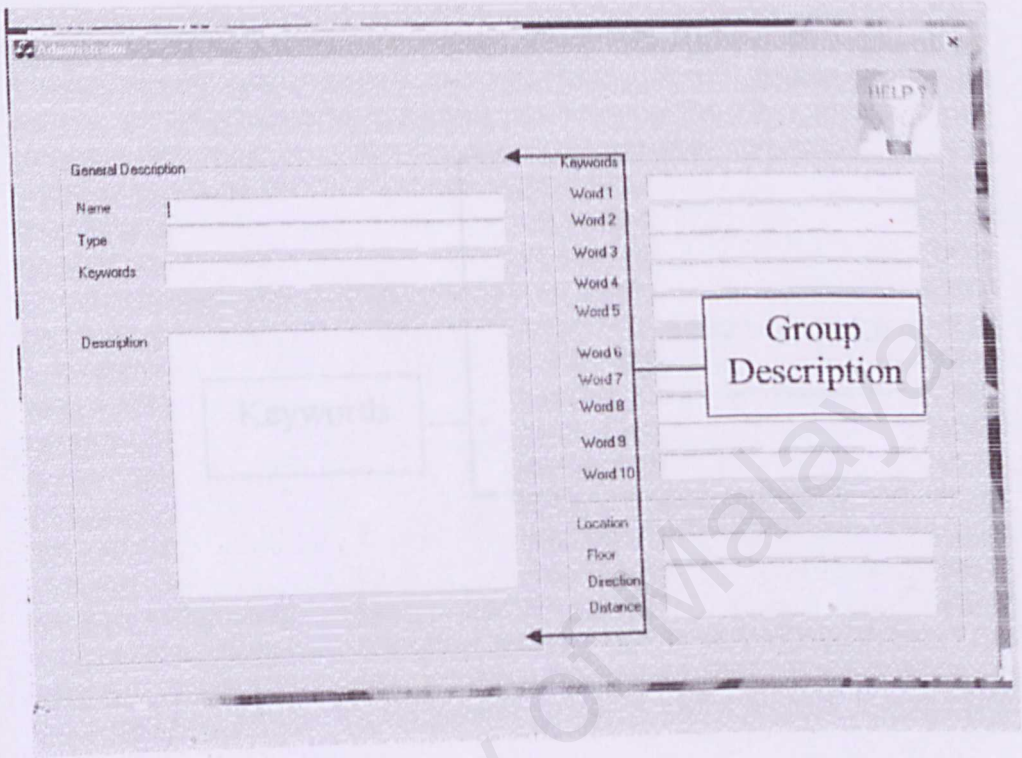

Figure 4 : General Description

The first group is the General Description, which, as the name itself suggests, provide a basic description of the information. There are small text fields and a large text field. They are:

1. Name : Name of the place \ item
2. Type : What type it is. E.g. shop, amenities, etc.
3. Keywords : A basic keywords describing it
4. Description : A small prose explaining what it is.

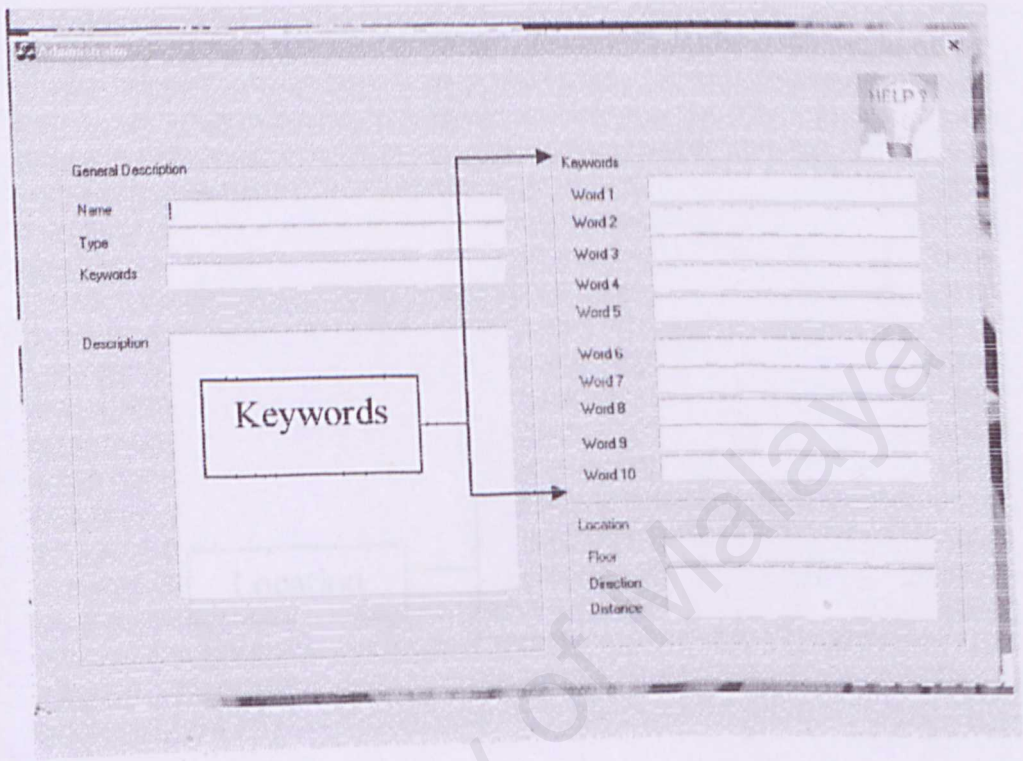

Figure 5 : Keywords

The second group is the Keywords. This is the keywords where the system will use to search for relevant information. Up to ten words can be entered. This enable the item to have relevant but not obvious keywords.

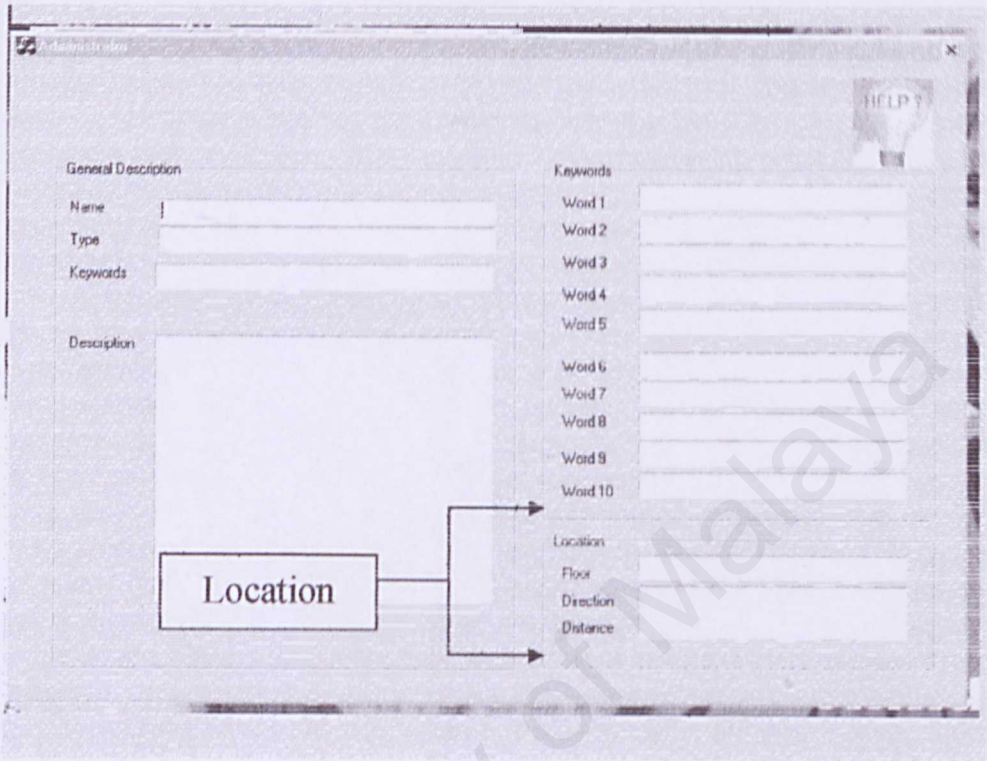

Figure 6 : Location

The location field will be used to determine the location of the item. Note that this location is relative to the point of origin, which will be determined earlier. There are:

1. Floor : Which floor it is
2. Direction : North, South, etc.
3. Distance : in meter from the origin (horizontally if applicable)